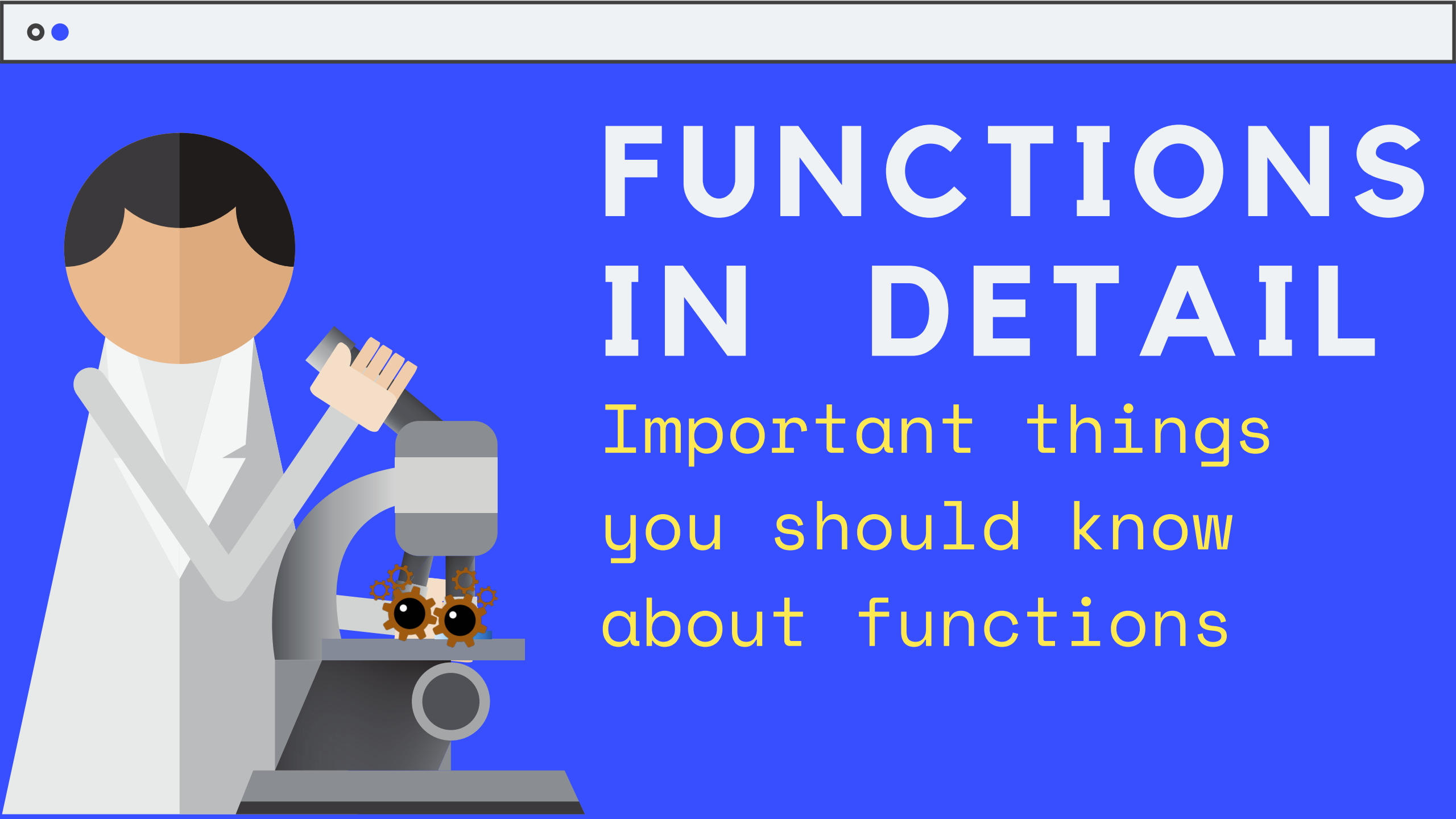




FUNCTIONS IN DETAIL

Important things
you should know
about functions

GOALS

- Understand Scope
- Write Higher Order Functions
- Pass functions as callbacks

SCOPE



Variable "visibility"

- The location where a variable is defined dictates **where we have access to that variable**



FUNCTION SCOPE

```
function helpMe(){  
    let msg = "I'm on fire!";  
    msg; //"I'm on fire";  
}
```

msg is scoped to the
helpMe function

```
msg; //NOT DEFINED!
```

FUNCTION SCOPE

```
let bird = 'mandarin duck';  
  
function birdWatch(){  
    let bird = 'golden pheasant';  
    bird; //'golden pheasant'  
}  
  
bird; //'mandarin duck'
```

bird is scoped to
birdWatch function

BLOCK SCOPE

```
let radius = 8;
```

```
if(radius > 0){
```

```
    const PI = 3.14;
```

```
    let circ = 2 * PI * radius;
```

```
}
```

```
console.log(radius); //8
```

```
console.log(PI); //NOT DEFINED
```

```
console.log(circ); //NOT DEFINED
```

PI & circ are
scoped to the
BLOCK

LEXICAL SCOPE

```
function outer() {  
  let hero = "Black Panther";  
  
  function inner() {  
    let cryForHelp = `${hero}, please save me!`  
    console.log(cryForHelp);  
  }  
  
  inner();  
}
```

FUNCTION EXPRESSIONS

There's another syntax we can use to define functions:

```
const square = function (num) {  
  return num * num;  
}  
square(7); //49
```



**FUNCTIONS
ARE...
OBJECTS!**



HIGHER ORDER FUNCTIONS





HIGHER ORDER FUNCTIONS

Functions that operate on/with other functions. They can:

- Accept other functions as arguments
- Return a function

FUNCTIONS AS ARGUMENTS

```
function callTwice(func) {  
    func();  
    func();  
}  
  
function laugh() {  
    console.log("HAHAHAHAHAHAHAHAHAHAHAHA");  
}  
callTwice(laugh) //pass a function as an arg!  
// "HAHAHAHAHAHAHAHAHAHAHAHAHAHAHAHA"  
// "HAHAHAHAHAHAHAHAHAHAHAHAHAHAHAHA"
```

RETURNING FUNCTIONS

```
function makeBetweenFunc(min, max) {  
  return function (val) {  
    return val >= min && val <= max;  
  }  
}  
  
const inAgeRange = makeBetweenFunc(18, 100);  
  
inAgeRange(17); //false  
inAgeRange(68); //true
```



CALLBACK FUNCTIONS



CALLBACK FUNCTIONS

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function.

CALLBACKS

**Callback
Function**

```
function callTwice(func) {  
  func();  
  func();  
}  
  
function laugh() {  
  console.log("HAHAHAHAHAHAHAHAHAHAHAHA");  
}  
callTwice(laugh) //pass a function as an arg!  
// "HAHAHAHAHAHAHAHAHAHAHAHA"  
// "HAHAHAHAHAHAHAHAHAHAHAHA"
```



HOISTING

Something I need to cover
that you shouldn't stress
about if it confuses you.

